Minority class augmentation using GANs to improve the detection of anomalies in critical operations

Waleed Hilal*[a], Connor Wilkinson[a], Alessandro Giuliano[a], Naseem Alsadi[a], Onur Surucu[a], Stephen A. Gadsden[a], John Yawney[b]
[a]McMaster University, 1280 Main St. W, Hamilton, ON, CA L8S 4L8;
[b]Adastra Corporation, 200 Bay St., Toronto, ON, CA M5J 2J2

ABSTRACT

With the ever-increasing adoption of interconnected technologies and rapid digitization observed in modern-day life, many online networks and applications face constant threats to the security and integrity of their operations or services. For example, fraudsters and malicious entities are continuously evolving their techniques and approaches to bypass current measures in place to prevent financial fraud, vandalism in online knowledge bases and social networks like Wikipedia, and malicious cyber-attacks. As such, many of the supervised models proposed to detect these malicious actions face degradations in detection performance and are rendered obsolete over time. Furthermore, fraudulent or anomalous data representing these attacks are often scarce or very difficult to access, which further restricts the performance of supervised models. Generative adversarial networks (GANs) are a relatively new class of generative models that rely on unsupervised learning. Moreover, they have proven to effectively replicate the distributions of real data provided to them. These models can generate synthetic data with a degree of quality such that their resemblance to real data is almost indistinguishable, as demonstrated in image and video applications – like with the rise of DeepFakes. Based on the success of GANs in applications involving image-based data, this study examines the performance of several different GAN architectures as an oversampling technique to address the data imbalance issue in credit card fraud data. A comparative analysis is presented in this paper of different types of GANs used to fabricate training data for a classification model, and their impact on the performance of said classifier. Furthermore, we demonstrate that it is possible to achieve greater detection performance using GANs as an oversampling approach in imbalanced data problems.

Keywords: Anomaly detection, generative adversarial networks, semi-supervised learning, machine learning, artificial intelligence, classification, imbalanced data

1. INTRODUCTION

Supervised anomaly detection techniques work under the assumption that the data set used consists of labelled instances that fall under either a normal or anomalous class. Most approaches under this category construct a predictive model for the normal and anomalous classes, and new unseen data can be classified by comparing it against the determined model. A significant issue with supervised anomaly detection, is that the anomalous class is usually rare in occurrence compared to the normal class [1]. It is also challenging to obtain accurate labels representative of the anomalous class, known as the class imbalance problem. In practical applications, the ratio between the classes can be as drastic as 1:10,000 [2].

Many techniques have been proposed in the literature to handle the imbalanced data issue, in both the algorithmic and data levels [3] [4]. In the former, algorithms or models are adjusted so that the bias towards the majority class is reduced to improve classification performance. Whereas in the latter, sampling techniques generate new samples in the minority class or eliminate samples from the majority class to balance the dataset. This class imbalance issue is encountered in binary and multi-class classification problems. However, in this paper, we focus on the binary problem and apply it to credit card fraud detection.

This study proposes using generative adversarial networks (GANs) as an oversampling strategy. Specifically, minority data instances are generated using the trained generator of a GAN and augmented into the training set of a classifier. GANs can parallelize sample generation with classification and avoid making assumptions about distributional and variational bounds compared to other oversampling techniques. Furthermore, GANs do not rely on Markov chains or maximum likelihood estimation. By designing and implementing various GAN architectures for oversampling fraudulent credit card

Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications IV,
edited by Tien Pham, Latasha Solomon, Proc. of SPIE Vol. 12113, 121131U
 · 0277-786X · doi: 10.1117/12.2618858

data, we present a comparative analysis of the performance of several different GAN architectures for generating data to improve the performance of a classifier. Some examples include the traditional vanilla GAN, the Wasserstein GAN (WGAN) and the Wasserstein GAN with gradient penalty (WGAN-GP). In section 2, we provide a brief background on the different GAN architectures examined in this paper and the classifier involved in the experiment. We follow with a discussion and overview of the proposed approach's experimental methodology and provide details on the dataset used in section 3. The experimental results are then presented in section 4, along with commentary and discussion of the relevant findings of the study. Finally, concluding remarks and suggestions for future research are given in section 5.

2. BACKGROUND

2.1 Generative Adversarial Networks

GANs are a type of deep learning framework proposed by Goodfellow et al. in 2014 that consists of two networks in competition with each other [5]. The first is a generative model G to capture the distribution of the training data. Its adversary is a discriminative model D that determines the probability of a sample coming from the training data instead of G. The objective in training G is to maximize the chance of inducing D, which is simply a classifier, to make mistakes distinguishing the data [5]. The two models in a GAN are deep learning architectures that learn a representation of the original input. Increasing the number of layers or the size of layers in the network can help it learn deeper and more abstract representations [6].

As illustrated in Figure 1, the generative model's input is random noise z, which it then transforms with a function and then produces examples of the real data [7]. The discriminator then learns to better distinguish between the real and generated examples by minimizing its prediction errors, and the generator tries maximizing the error, resulting in a competition formalized as a minimax game in (2.1):

$$\min_{\theta_G} \max_{\theta_D} (\mathrm{E}_{x\sim p_d}[\log D(x)] + \mathrm{E}_{z\sim p_z}[\log \left(1 - D\big(G(z)\big)\right)]) \tag{2.1}$$

where θ_G and θ_D are the parameters of the generator and discriminator networks, respectively, p_d is the data distribution and p_z is the prior distribution of the generative network [5]. Although GANs are unsupervised learning algorithms, they use a supervised loss as part of the training. In most financial fraud applications to date, GANs have been used in a semi-supervised fashion as a method of oversampling for data augmentation, which is the reasoning behind their classification under semi-supervised in this paper [7].

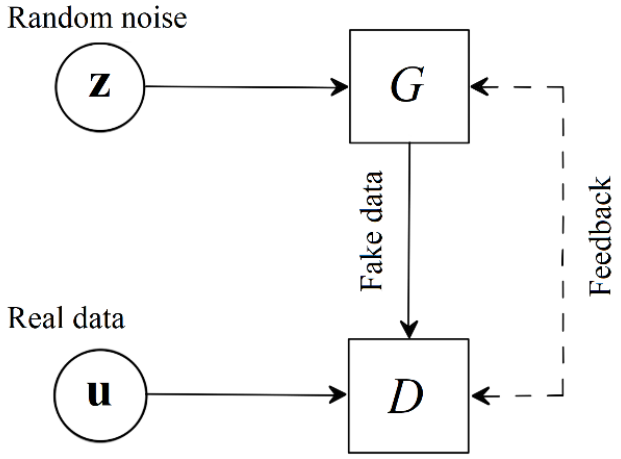

Figure 1. Schematic of a GAN's generator *G*, accepting random noise *z* as input and outputting generated examples to the discriminator *D*. The discriminator distinguishes the generated examples by *G* from the real data *u*.

2.2 Wasserstein GANs

Traditional GANs often face the problems of mode collapse and vanishing gradients. Mode collapse is when the GAN's generator or discriminator collapses to one or a few modes of those present in the entire distribution of the data, with the remainder of the modes disappearing. This is an issue as real-world datasets often have many modes associated with each different class. Furthermore, the vanishing gradient problem is associated with the loss function of traditional GANs. This issue arises during the training stage, specifically when the discriminator is getting better and better, to the point where the feedback it provides becomes less informative to the generator. In fact, the discriminator may result in gradients close to zero, which is not helpful to the generator, and thus, it is unaware of how it can improve.

The Wasserstein GAN (WGAN) has been proposed by Arjovsky *et al.* in [8] as a means of addressing the issues associated with traditional GANs. The WGAN makes use of a modified loss function, which is based on a measure known as the Earth Mover's (EM) distance. The EM distance measures how different two distributions are by estimating the amount of effort it would take to make the generated distribution equal to the real. The function depends on both the distance and the amount of the generated distribution. Unlike the traditional GAN, it does not have flat regions where the distributions start to get very different. Thus, the discriminator is prohibited from improving a lot, and as such, the issue of vanishing gradients is eliminated, while also reducing the likelihood of mode collapse.

With the WGAN, the discriminator is referred to as the critic, as denoted by C in (2.2), the modified loss function of a WGAN. In a traditional GAN, the output of the discriminator must be a prediction between 0 and 1. The Wasserstein loss, however, does not have that requirement. Instead, its output can be interpreted as how real the critic considers an image to be. The name change from discriminator to critic is due to the output no longer being bounded by 0 and 1. The critic is essentially responsible for maximizing the distance between its evaluations on a fake and its evaluations on a real sample.

$$\min_{\theta_G} \max_{\theta_C} \mathrm{E}_{x\sim p_c}[C(x)] - \mathrm{E}_{z\sim p_z}\big[C\big(G(z)\big)\big] \tag{2.2}$$

Training a WGAN using the Wasserstein loss requires the fulfilment of a special condition to prevent the generation of poor samples or failure to converge, which is being 1-Lipschitz continuous. Specifically, this means that the norm of the critic's gradients must be at most one. Originally, the authors of [8] suggested using a weight clipping method. However, this has proven to be computationally intensive. Instead, a WGAN with a gradient penalty (WGAN-GP) has been proposed as a more favourable alternative to ensure Lipschitz continuity. The WGAN-GP's loss function involves adding a regularization term to the loss function of a WGAN from (2.2). This regularization term penalizes the critic when the norm of its gradient is greater than one, as can be seen in (2.3) below [9]:

$$\min_{\theta_G} \max_{\theta_C} \mathrm{E}_{x\sim p_c}[C(x)] - \mathrm{E}_{z\sim p_z}\big[C\big(G(z)\big)\big] + \lambda \mathrm{E}_{\hat{x}\sim p_{\hat{x}}}[(\|\nabla_{\hat{x}} D(\hat{x})\|_2 - 1)^2] \tag{2.3}$$

where $\hat{x}\sim p_{\hat{x}}$ are random sample and λ is a hyperparameter value which must be tuned, signifying how much to weigh the regularization term against the loss function.

2.3 XGBoost

In this study, we propose using the popular eXtreme Gradient Boosting (XGBoost) classification model, which is based on the gradient-boosted decision tree algorithm (GBDT) [10]. In addition, XGBoost can be considered state-of-the-art in classification models, performing exceptionally well in detection accuracy while effectively using computational resources to be able to handle billions of samples with far fewer resources than traditional approaches.

The XGBoost model is composed of multiple regression trees, whereby the final output is the consequence of the additive combination of the decision results of all subtrees. This is known as an ensemble approach, where with a high number of individually weak but complementary classifiers, the resultant is a robust estimator. The term boosting refers to the nature in which new models are added to the ensemble sequentially, where at each particular iteration, a new weak, base-learner model is trained with respect to the error of the whole ensemble learnt so far [11]. Thus with XGBoost, the principle idea is to construct the new base-learners to be maximally correlated with the negative gradient of the loss function associated with the whole ensemble [11].

3. METHODOLOGY

As discussed, in this paper, we aim to showcase how GANs can be used as an method to handle data oversampling. Oversampling implies that the the dataset being used has a heavy imbalance between classes. In this paper, we focus on a credit card fraud dataset, wherein the fraudulent data is heavily outweighed by the benign data. To rectify this imbalance, we created GAN models to virtually create more fraudulent data points to be used in the classifier's training. The data used is from the credit card fraud detection dataset by ULB Machine Learning Group, collected in September 2013 from European cardholders. The dataset is composed of 30 features, including time and amount, and other features which have been transformed using principal component analysis (PCA) to protect users' privacy. The dataset is labelled with a binary indicator to identify between fraudulent and non-fraudulent data. As mentioned, there is a high imbalance between classes, with only 492 fraudulent samples of a total sample size of 284,807 samples, meaning the fraudulent data makes up only 0.17 percent of the total samples.

3.1 Data Preprocessing

Prior to any training of the models, the dataset involved in the study was examined in order to eliminate any issues associated with improper or unclean data. The first observation from the data, was that the transaction values were highly skewed to the left, with most transactions having a value less than 2,500, and some transactions reaching a value as high as 25,000. This original distribution is shown in Figure 2. In order to account for the low number of extremely high transaction values, this feature was log-transformed, resulting in a near-normal distribution, as can be seen in Figure 3.

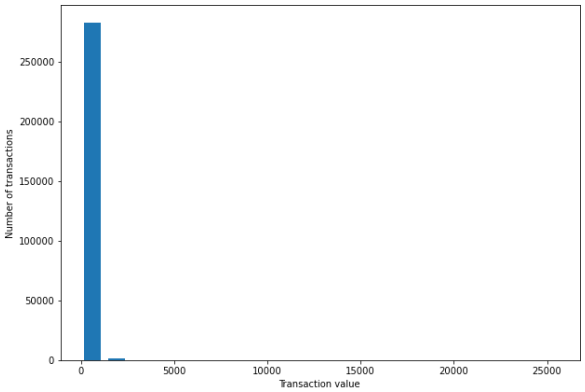

Figure 2. Original distribution of transaction values in the credit card fraud dataset.

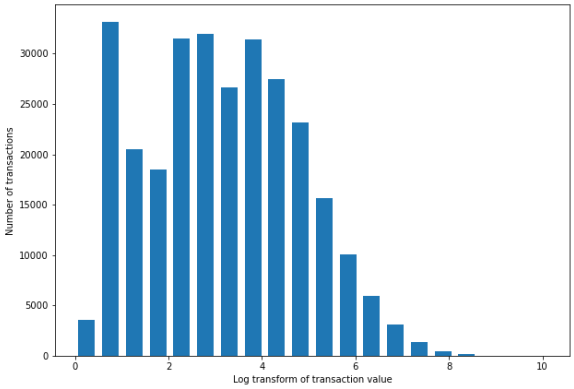

Figure 3. Distribution of log-transformed transaction values

Additionally, we noticed that the time feature was expressed in seconds format, spanning a period of two days. To allow for more interpretable outcomes and representation of behaviors over the span of those two days, the time feature was transformed from seconds to hours. As such, transactions can occur from 0^{th} hour (start of the data), up to the 48^{th} hour (end of the data). Besides the aforementioned time and transaction amount features, all other features were already affected by PCA, and were consequently not altered.

3.2 Network Structure and Training

The proposed network structures of the GAN, WGAN, and WGAN-GP follow similar design and training procedures. Firstly, each architecture's generator and discriminator are modelled as a multilayered perceptron (MLP), otherwise known as an artificial neural network (ANN). Both the generator and discriminator were designed with 3 hidden layers, maintaining a rather simple structure, as well as an input and output layer. The generator's input layer consists of 32 perceptrons, which was an arbitrary choice for the random noise vector which acts as input to the generator. Subsequent hidden layers in the generator then double in size, and consist of 64, 128 and 256 layers in the first, third and hidden layer, respectively. This generic structure for the generator of each GAN is graphically shown in Figure 4. As for the discriminator, its' structure is symmetric to that of the generator, in that the input layer takes the input data, which consists of the output of the generator, and passes it to three subsequent hidden layers with 256, 128 and 64 perceptrons before it classifies each generated sample as real or fake.

In the hidden layers of the generators, a leaky rectified linear unit (ReLU) function is used with the hyperparameter alpha tuned to a value of 0.2. We also employ batch normalization in each of the layers, as it has been observed to improve stability during the training phase. Similarly, the discriminator's perceptrons involved a leaky ReLU function with an alpha value of 0.2. In addition to this, a dropout chance of 15% was incorporated for each perceptron in the discriminator, rather than batch normalization, which reduced or eliminated the chance of overfitting the model. A batch size of 128 was used throughout the training stage. Furthermore, it is important to note that the activation function of the penultimate layer of the discriminator uses the sigmoid function to generate a binary classification. In the WGAN and WGAN-GP, however, this sigmoid function is not used, as the output of the critic is not bounded between 0 and 1, as previously described in section 2 of this paper. The choice of hyperparameter values for the networks, such as the learning rate, dropout probability, alpha parameter, size of each layer and choice of activation function were determined through an extensive grid search.

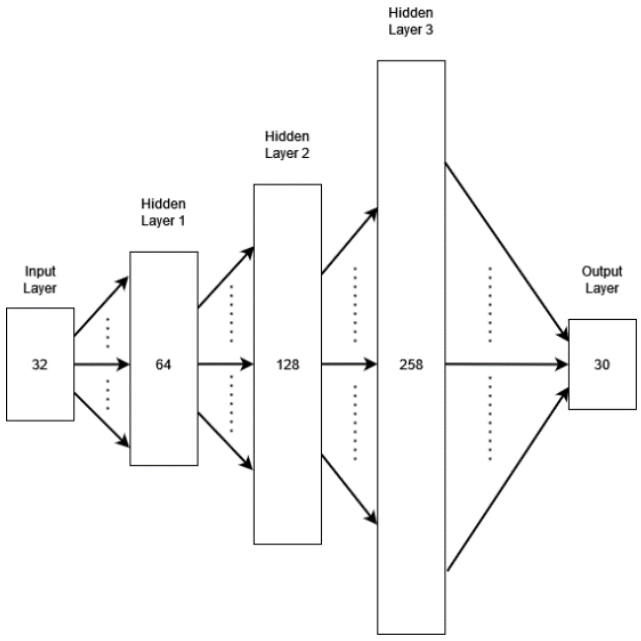

Figure 4. Representation of the network structure of the generators of all the different GANs proposed in this study, which are the same regardless of the type of GAN used.

3.3 Data Augmentation and Classification

Following successful training of the GAN architectures, an XGBoost model was designed for the binary classification task of detecting fraudulent credit card transactions. The training dataset's minority class was oversampled using random oversampling (ROS), and this was used to tune the hyperparameters of the baseline classification model through an extensive grid search. The tuned hyperparameters of the baseline classification model were then utilized for the subsequent models involving data generated from the GAN, WGAN and WGAN-GP.

Each of the generators from the trained GAN architectures were made to produce a set number of fraudulent samples, which was chosen to be 227,057. The number of samples generated was based on the difference of legitimate and fraudulent transactions in the training set, and with the goal of ensuring a balanced dataset was used for the training of the classification model. Subsequently, each respective classification model augmented with GAN-generated fraud data is evaluated on the test set. The following metrics are then used to compare the performance of each model:

$$accuracy = \frac{1}{n}\left(\sum_{i=1}^{n}|\hat{y}_i - y_i|\right) \tag{3.1}$$

$$precision = \frac{TP}{TP + FP} \tag{3.2}$$

$$recall = \frac{TP}{TP + FN} \tag{3.3}$$

where, $\hat{y}_i$ is the predicted value, y_i is the true value, TP is the number of true positives predicted, FP is the number of false positives predicted, and FN is the number of false negatives predicted. Note that each of these performance measures is bounded between 0 and 1.

4. RESULTS AND DISCUSSION

The generated fraudulent sample for each model was compared against the fraudulent samples from the original dataset. For each set of real and generated data samples, a boxplot was produced highlighting the distribution of each feature in each dataset. Figure 5 below shows the distribution of the samples generated by the vanilla GAN. It can be seen from this figure that the vanilla GAN struggles to generate samples that capture the entire distribution of the actual fraud data. Instead, the vanilla GAN can be seen to encounter the issue of mode collapse, whereby the generator only learns to produce one or a very few modes of the original distribution.

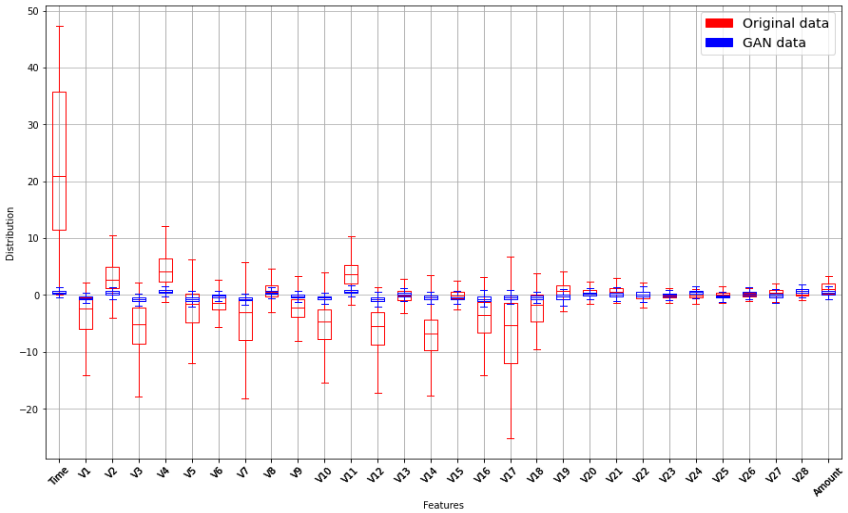

Figure 5. Boxplot of the real fraudulent data (red) and the fake fraudulent data (blue) generated by a GAN.

In Figure 6, the boxplot of the data generated by a WGAN is displayed. It is evident from this figure that the distribution of the data produced by the WGAN's generator more closely resembles that of the original data than the GAN from Figure 5. Furthermore, it is clear that the mode collapse issue has been somewhat eliminated and the WGAN is able to capture more modes and a wider representation of the original data. However, it can be observed that for certain features, the WGAN either overshoots or undershoots the interquartile range (IQR) of the features of the original data. Furthermore, despite having improved stability, it can still be seen that some features exhibit some form of mild mode collapse.

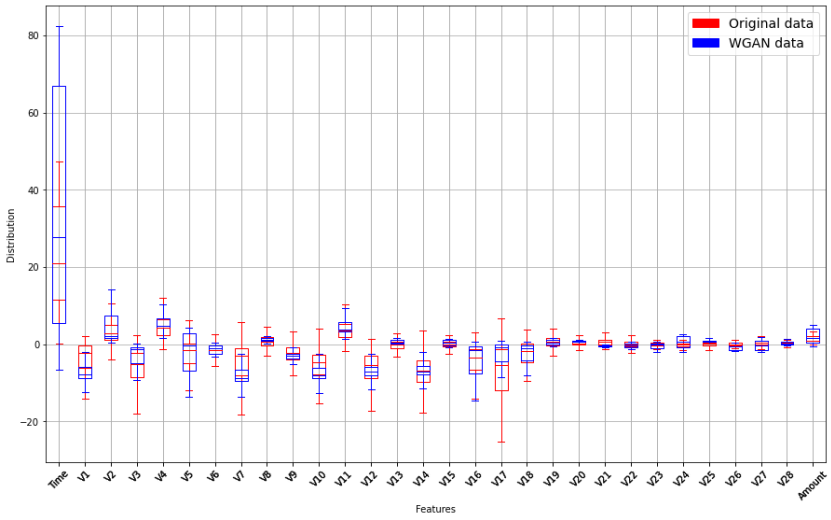

Figure 6. Boxplot of the real fraudulent data (red) and the fake fraudulent data (blue) generated by a WGAN.

Finally, it can be seen from Figure 7 that the distribution of the data generated by the WGAN-GP far exceeds that of the previous two models in terms of accurately matching the distribution of the original fraudulent data. The IQR of each of the features generated by the WGAN-GP almost perfectly track that of the original, except for slight overestimates and underestimates of the 25th and 75th percentile of some features. This under or overestimation is most visible in the time feature, where it is clear that the latter of the two occurs – otherwise, the second most obvious case is the overestimation of the 25th percentile in the 'V17' feature.

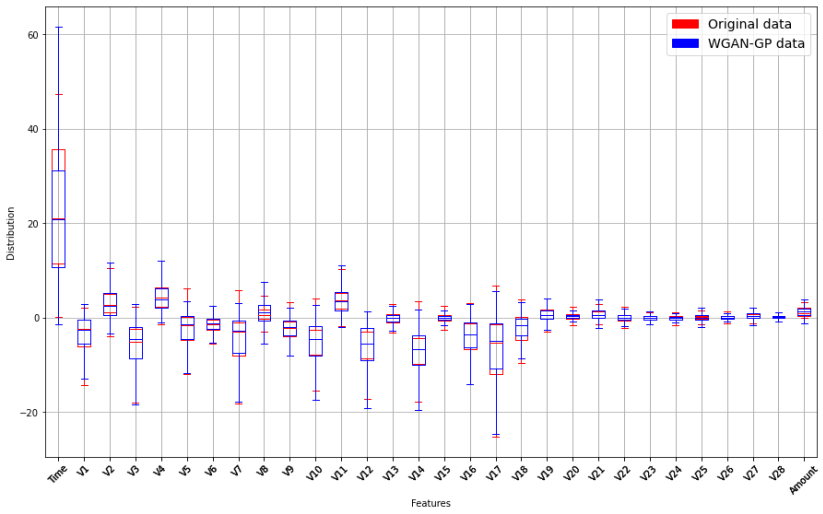

Figure 7. Boxplot of the real fraudulent data (red) and the fake fraudulent data (blue) generated by a WGAN-GP.

Table I. Distribution of class labels in the dataset explored

	ROS	GAN	WGAN	WGAN-GP
Accuracy	0.999508444	0.999526000	**0.999543555**	0.999438222
Precision	0.872340426	**0.917647059**	0.909090909	0.817307692
Recall	0.836734694	0.795918367	0.816326531	**0.867346939**
F1 score	0.854166667	0.852459016	**0.860215054**	0.841584158
AUROC	0.918261832	0.897897633	0.908092922	**0.933506404**

Following the augmentation of the generated data from each GAN architecture to the respective XGBoost models, the classification performance measures for each was recorded, as can be seen in Table I. In this table, the highest score achieved by a model for each of the performance measures is indicated in bold text. From initial examination of Table I, the models augmented with GAN-generated data result in superior performance in terms of accuracy, precision, recall, F1 and AUROC scores than ROS.

Specifically, it can be seen that the vanilla GAN results in a 5.2 percent increase in terms of precision compared to ROS, with negligible change in accuracy and F1 score. However, this increase in precision comes at the cost of a 4.9 percent decrease in recall and a 2.2 percent decrease in AUROC. Since this study involves the task of credit card fraud detection, these results are not entirely favourable as the costs associated with a lower recall are greater than that of an improved precision. The WGAN's results indicate that it addresses the issues associated with the vanilla GAN in terms of the decreases in F1 score and AUROC. Furthermore, it is apparent that the WGAN achieves the highest accuracy and F1 score, at the cost of a 1 percent decrease in precision from the vanilla GAN model. However, this is not enough to warrant the WGAN superior to ROS, as the recall remains inferior by 2.4 percent.

Finally, it is apparent that the WGAN-GP results in the greatest recall score of all the models examined from Table I, with an increase of approximately 3.7 percent compared to that of ROS. This signifies that of all the models, the WGAN-GP is able to detect more fraudulent transactions than all other GAN architectures, and even ROS. This increase in recall is also accompanied with the highest AUROC achieved by the WGAN-GP of any of the models, improving by 1.7 percent compared to ROS. This improved performance comes at the cost of a non-trivial decrease in precision, by up to 6.3 percent. As such, it can be said that the WGAN-GP's performance is favourable over other GAN architectures and ROS for imbalanced data problems, assuming an increase in the amount of false positives is tolerable. It can also be inferred that regardless of the drop in precision, the WGAN-GP is the best overall architecture due to having the greatest AUROC score of all the architectures studied in this paper.

5. CONCLUSION

In this study, a comparative analysis of different GAN architectures applied to augmenting minority class samples into an unbalanced training set was carried out. Namely, a vanilla GAN, WGAN and WGAN-GP were designed, tuned and implemented to produce artificial data resembling fraudulent credit card transactions, so that when augmented into the training set of an XGBoost classification model, improved detection of the fraudulent class would be achieved. It was demonstrated throughout this study that GANs often face the issue of mode collapse and vanishing gradients, resulting in instability during training and the generation of unsatisfactory low-spectrum samples. By adding a gradient penalty to the WGAN architecture, it was proven that the result is a more encompassing distribution by the generated data, which much more closely resembles that of the original data. Furthermore, when augmented with samples generated by a WGAN-GP, it has been further corroborated that a classification model's ability to detect fraudulent instances is improved. This improvement, however, comes at the cost of a slight increase in the amount of false alarms by the model. Thus, the assumption must be made that the benefit associated with the increased detection of fraud outweighs the cost associated with more false positives. Future work involves exploring more complex methods, such as time series forecasting models

like long short-term memory (LSTM) networks as the generator or discriminator of a GAN to account for the temporal behaviour of the nature of data used in this study.

REFERENCES

[1] C. Phua, D. Alahakoon and V. Lee, "Minority report in fraud detection: classification of skewed data," *ACM SIGKDD Explorations Newsletter,* vol. 6, no. 1, pp. 50-59, June 2004.

[2] N. V. Chawla, N. Japkowicz and A. Kotcz, "Editorial: special issue on learning from imbalanced data sets," *ACM SIGKDD Explorations Newsletter,* vol. 6, no. 1, pp. 1-6, 2004.

[3] N. V. Chawla, K. W. Bowyer, L. O. Hall and W. P. Kegelmeyer, "SMOTE: Synthetic Minority Over-sampling Technique," *Journal of Artificial Intelligence Research,* vol. 16, pp. 321-357, 2002.

[4] H. He, Y. Bai, E. A. Garcia and S. Li, "ADASYN: Adaptive synthetic sampling approach for imbalanced learning," in *IEEE International Joint Conference on Neural Networks,* Hong Kong, China, 2008.

[5] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville and Y. Bengio, "Generative Adversarial Nets," *Advances in Neural Information Processing Systems,* vol. 27, pp. 2672-2680, 2014.

[6] Y. Bengio, A. Courville and P. Vincent, "Representation Learning: A Review and New Perspectives," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 35, no. 8, pp. 1798-1828, 2013.

[7] W. Hilal, S. A. Gadsden and J. Yawney, "Financial Fraud: A Review of Anomaly Detection Techniques and Recent Advances," *Expert Systems with Applications,* vol. 193, p. 116249, 2022.

[8] M. Arjovsky, S. Chintala and L. Bottou, "Wasserstein GAN," 2017. [Online]. Available: arXiv:1701.07875 .

[9] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin and A. Courville, "Improved Training of Wasserstein GANs," 2017. [Online]. Available: arXiv:1704.00028v3.

[10] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining,* San Francisco, CA, 2016.

[11] A. Natekin and A. Knoll, "Gradient boosting machines, a tutorial," *Frontiers in Neurorobotics,* vol. 7, p. 21, 2013.